INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 3, PP. 715–726
Manuscript received December 10, 2023; revised July, 2024. doi: 10.24425/ijet.2024.149601

A new efficient predictor blending lossless image coder

Grzegorz Ulacha, and Ryszard Stasinski

***Abstract*—In the paper a highly efficient algorithm for lossless image coding is described. The algorithm is a predictor blending one, a sample estimate is computed as a weighted sum of estimates given by subpredictors, here 27 ones, hence the name Blend-27. Data compaction performance of Blend-27 is compared to that of numerous other lossless image coding algorithms, including the best currently existing ones. The compared methods are "classical" ones, as well as those based on Artificial Neural Networks. Performance of Blend-27 as a near-lossless coder is also evaluated. Its computational complexity is lower than that of majority of its direct competitors. The new algorithm appears to be currently the most efficient technique for lossless coding of natural images.**

***Keywords*—image coding; lossless coding; predictor blending**

I. Introduction

DESPITE its restricted range of applications lossless image coding is an important part of image processing. It is used for medical and forensic data compression, in geologic, astronomic, and space imaging, finally in professional photography and video (compression of RAW, TIFF, and some other file types). The most popular methods are standardized by JPEG, [1], [2], also the new WebP [3], nevertheless, they are time-efficiency oriented, hence, do not address an interesting question: how much redundancy can be shaved-off uncoded image files? The first still regarded as highly efficient algorithm was introduced in 1996 CALIC [4]. As computer performance is growing steadily, today its complexity can be described as low, and its place as coding efficiency yardstick is taken by much better, but much more complex approaches: TMW method (1997) [5] and its later expansion TMW$^{\mathrm{LEGO}}$ (2001) [6], WAVE-WLS (2002) [7], MRP 0.5 (2005) [8] and its newer extended versions, xMRP [9] (2008), MDL-PAR [10] (2011), GPR-BP [11] (2014), and MRP-SSP [12] (2014), finally, by PMO [13], [14], [15] (2018, 2019, 2022), and EM-WLS [16] (2020) algorithms. Between these extremes there is a handful of intermediate methods, both in terms of coding performance and complexity: CoBALP$_{\max}$ [17], GLICBAWLS [18], SWAP [19], and many others. Another interesting group of algorithms ranging from simple to the most advanced ones are based on predictor blending, the first and simplest being Blend-7 [7], then an advanced Blend-20 [20].

An emerging technology in the area are coders based on artificial neural networks (ANNs). Training an ANN is a slowly advancing task, hence, intrinsically such coders have relatively high computational complexity. However, as ANNs became an important tool of artificial intelligence, a great effort is put on computing software and architectures reducing ANN training cost, e.g. there are applications delegating it to a GPU. This encourages research in the area. First ANN-based codecs are trained on coded image samples, and work as backward predictors [19], [21], [22], [23], [24]. When run on CPUs such coders have relatively high computational complexity, while obtained results are average, in the case of the best one worse than for WLS method [24]. Newer codecs are usually trained on sets of exemplary images, and some results are remarkable [25], [26], section IV-B in this paper. An analysis of deep learning algorithms advantages and drawbacks can be found in [27], [28].

In the paper a signalized in [29] new powerful image lossless coding method is presented named Blend-27. Data modelling part of the coder is described, description of the highly sophisticated arithmetic coder can be found in [16]. Section I-A provides indexing convention used in the prediction formulae, section I-B introduces the near-lossless coding idea. Then adaptive lossless coding techniques are presented, section II: forward and backward adaptation methods, sections II-A and II-B, and cascade systems, section II-C. The algorithm is presented in Section III, sub-sections contain: general characteristic of used sub-predictors, section III-A, and blending formulae, section III-B. Experiments are reported in section IV, firstly, a note about algorithms computational

The paper is partly supported by Polish Ministery of Science and Education grant 0314/SBAD/0241.

G. Ulacha is with Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Szczecin, Poland (e-mail: gulacha@wi.zut.edu.pl).

R. Stasinski is with Institute of Multimedia Telecommunications, Poznan University of Technology, Poznan, Poland (e-mail: ryszard.stasinski@put.poznan.pl).

			46	42	38	43			
		37	32	26	24	27	33	39	
	36	29	20	16	14	17	21	30	40
45	31	19	11	8	6	9	12	22	34
41	25	15	7	3	2	4	10	18	28	44
35	23	13	5	1	x_n				

Fig. 1. Indexes of samples in the neighborhood of coded pixel, or error sample.

complexity is given in section IV-A, followed by comments on lossless and near-lossless data compression performance of few dozens methods, sections IV-B, IV-C, Tables II, III, IV, V, VI, VII, and VIII, Figure 5. As can be seen, Blend-27 seems to be currently the most efficient lossless image coding algorithm.

A. Indexing of two-dimensional data

Modern lossless compression methods usually consist of two stages: data modeling, and actual compression using an entropy coder. In both stages it is necessary to define somehow the neighborhood of the currently coded pixel, or its processed value, or its estimation error. In Figure 1 used in this paper indexing scheme for neighborhoods of a pixel or coded prediction error is presented. The pixel of interest is denoted by $x_{(n)} = P(0)$, others are pointed out by indices. For example, if the sample $P(4)$ forms a sub-predictor for the currently coded one, Table I, then the sub-predictor error is obtained by subtracting value of pixel on position "$n+4$" from that on position "n". Similarly, for a linear predictor of rank r:

$$\hat{x}_{(n)} = \sum_{j=1}^{r} b_j \cdot P(j) \tag{1}$$

where b_j are predictor coefficients, summed up pixels are taken from positions shown in Fig.1. In general, the index value is obtained by taking into account two rules: minimization of Euclidean distance, and clockwise ordering of samples having the same Euclidean distance from sample of interest. The same indexing scheme is used for neighborhood prediction errors, $e(0) = e_{(n)}$:

$$e_{(n)} = x_{(n)} - [\hat{x}_{(n)}] \tag{2}$$

where predictor output $\hat{x}_{(n)}$ is rounded up. In order to simplify notation in the following text lower indices $_{(n)}$ are omitted.

B. Near-lossless coding

Lossy compression methods lead to much higher compression ratios than lossless ones. The reconstructed images often look "like original" ones, still in strongly variable image areas some tiny details can be distorted, or even missing. This observation leads to near-lossless methods in which maximum difference between pixels of coded and original images d is imposed [18], [30], [31], [32], [33], [34]. Entropy coding is done for quantized prediction error e (2), [34]:

$$\hat{e} = \begin{cases} \lfloor \frac{e+d}{2d+1} \rfloor, & \text{for} \quad e \geq 0 \\ \lfloor \frac{e-d}{2d+1} \rfloor, & \text{for} \quad e < 0 \end{cases} \tag{3}$$

The approximate value of prediction error can be reconstructed:

$$e \approx \hat{e} \cdot (2d+1). \tag{4}$$

The difference d is determined by experts, moreover, Regions of Interest (ROI) in an image can be coded with $d = 0$, i.e. losslessly [35].

II. Adaptive predictive techniques

The simplest predictor coders are fixed ones, usually with few coefficients equal to simple combinations of powers of 2, e.g. $0.625 = 0.5 + 0.125$. Multiplications by such coefficients can be replaced by additions and bit shifts [36], hence, realizations of such coders are very simple.

Nevertheless, such coders are rather inefficient, even the simplest coders used in practice have some signal adaptation mechanism. Image coder parameters can be determined once for the whole image (by a forward adaptation method) or individually for each coded pixel (by a backward adaptation technique). The advantage of backward adaptation is the possibility of using relatively high prediction orders (there is no need to provide coder parameters to the decoder), which allows for high compression efficiency. However, the disadvantage of this solution is the necessity of updating prediction coefficients in both the encoder and decoder for each pixel (it is a time-symmetric approach). The forward adaptation decoders can be much simpler than coders. The approaches lead to different coder construction philosophies.

A. Forward adaptation methods

The methods start with analysis of image properties, global or local. The analysis determines parameters of data modelling stage, the parameters should be sent to the decoder in a header. Then, there is a trade-off between complexity of signal model (header size) and coder efficiency. Relatively good performance and short headers are obtained for Minimum Mean Square Error (MMSE) optimized single global predictors, for example for predictor length $r = 24$ and 11 bits for coding a predictor coefficient the header for a 512×512-pixel image increases coder bit rate by 0.00097 bits per pixel [16], [37].

Two classical image lossless techniques, JPEG-LS [1], and CALIC [4], use fixed local predictors: one chosen from 3 ones by Median Edge Detector (MED), and one from 7 ones by Gradient-Adjusted Predictor (GAP) approach, respectively. MED and GAP are determining *context* for a coded pixel. The techniques represent very good trade-off between coder complexity and efficiency. For both MED and GAP methods exist their improved versions: e.g. MED_+ [38], GAP_+ [39],

and GAP version in which predictors are optimized for coded image [40]. The context approach is quite popular: [41], [42], [43], [44], [45], [46], [47], [48], including its generalization allowing introduction of very large number of contexts (e.g. 2048) [49].

Another local analysis approaches consists in constructing signal models for image blocks (e.g. 8×8 or 16×16 pixels). One of the first solutions of this type is the method presented in paper [50], where to each block of 8×8 pixels is assigned one of 8 fixed models, giving the smallest absolute error. In this way the header information associated with a block require only three bits forming the model number.

More advanced methods use the MMSE criterion for determining the best prediction coefficients. Unfortunately, this is associated with very large header information, as predictor coefficients require large numbers of bits. To reduce the size of the header the blocks with similar characteristics are grouped into clusters, and all are associated with a common prediction model [5], [6], [51]. The use of vector quantization techniques (as well as fuzzy clustering [52]) leads to optimised sets of e.g. 16 prediction models, so that even for a large prediction order the header size does not increases the bit average significantly. In work [8] a technique of combining adjacent blocks belonging to the same category into groups is used (associated with the same predictor) to create larger blocks. Then map of the blocks having variable sizes is saved using an effective technique of encoding quadtrees. The described in [8] codec MRP 0.5 is predecessor of a line of the most advanced forward adaptation methods: xMRP [9], GPR-BP [11], MRP-SSP [12]. Multiple iterations of signal model parameters tuning make this techniques stronly time asymmetric (Much longer coding than decoding times).

In some works MMSE quantization criterion is replaced by some other one. In [53], [54], [55] genetic algorithms are used for predictor optimization. The approach is computationally complex, hence, the considered predictor ranks are $r = 3$, or $r = 4$, only. Much better results are obtained for generalizations of MMSE optimization criterion. In [56] a coder based on MRP 0.5 algorithm using Minimum Mean Absolute Error (MMAE) is presented. An analysis of application of such criterions to image coding can be found in [57].

B. Backward adaptation methods

In backward adaptation methods coder and decoder have access to the same image pixels, hence, decoder can properly reconstruct signal model without any information from the coder. This means that information about model and its complexity do not influence coded signal bit rate. This means that a decoder should be as complex as a coder, and the techniques are time symmetric. Another problem is an algorithm start, at the beginning there are not enough pixels to form full signal model. Even worse is the situation of context algorithms, as in their case several models should be initialized. This is particularly apparent in the case of low-resolution images.

The simplest backward adaptation algorithms are LMS and NLMS ones. Image signal is often strongly variable, despite this it can be treated as locally stationary. This means, however, that LMS convergence coefficient should be set to the worst case value being rather small [58]. This results in slow conversion rate and generally low performance of these methods. Moreover, existing formulae on convergence coefficients are optimized for one-dimensional signals. Summarizing, it is not surprising that the approach is rarely used.

Some improvement is obtained if a predictor is preceded by a transform [17], [59], or even better, context coding is performed and there are several predictors optimized for contexts. In the simplest case the predictors can be optimized using LMS or NLMS algorithms [60], [61], but better results are obtained for quickly converging RLS [62], or its "uncompromised" version OLS [63], [64], [65]. In the case of OLS the local approximation of signal autocorrelation matrix is computed in the window Q, used also in calculation of third and fourth penalty terms determining importance of Blend-27 subpredictors, Figure 3, section III-B. Further improvements of this approach can be found in [16], the best, but also the most complex algorithm presented there is EM-WLS, in which a mix of WLS models of different size is realized. The techniques in which predictor outputs are *blended* rather than *selected* are described in section III.

Recently emerged an interesting line of algorithms not using predictors, but evaluating probability models for an arithmetic coder, Probability Model Optimization (PMO) techniques [13], [14], [15]. As it is shown in section IV-B, they are the main competitors for the Blend-27 method, section IV-B.

A separate group of backward adaptation algorithms is based on Artificial Neural Networks (ANNs), in the first papers the networks learn on pixels of the coded image. The Adaptive Neural Networks (AdNN) are used in [19], [21], [22], the cellular ones in [23]. It is shown in [24] that more advanced context algorithm using AdNNs is somewhat worse than classical method, WLS, while significantly more computationally complex. The newer algorithms based on deep learning are even more complex, but possibility of using GPUs for speeding-up the learning process encourages research in this domain. There are algorithm in which ANN based coders support the work of the basic one [66], [67]. An important group form methods in which ANNs are initially trained on exemplary images, usually small ones, of size 32×32-, or 64×64-pixels [28], [68]. As observed in [15], [69], such small training images often result in unsatisfactory performance for images of practical resolution. The general opinion is that in practice codecs based on ANNs are still less efficient while more computationally complex than "classical" ones [27],

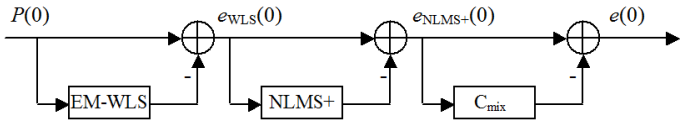

Fig. 2. Cascade predictor lossless image codecs, e.g. Extended Multi-WLS (EM-WLS) one [16], and some sub-predictors of Blend-ν algorithms, Table 1.

[28], nevertheless, performance of two codecs appear to be promising [25], [26], section IV-B.

C. Cascade systems

An idea implemented in newer lossles coding techniques is their cascade organization, the main predictor is followed by a stage or stages of NLMS (Normalized Least Mean Square) filters, Figure 2. This is the case of sub-predictors of some Blend-ν algorithms, [20], [29], but also the case of EM-WLS method [16]. Up to now the idea was mainly used in audio coding [70]. Additionaly, in the third cascade stage sub-predictor bias cancellation can be done, C_{mix} in Figure 2. Existence of such stages is pointed out in columns NLMS+ and C_{mix} in Table I. Used in Blend-27 the most advanced update formulas on NLMS coefficients can be found in [16], there are two NLMS sub-stages of rank $r_{NLMS1} = 106$ and $r_{NLMS2} = 42$.

It has been discovered by authors of CALIC that predictors tend to accumulate biases superimposed on their estimates. Then, both CALIC and JPEG-LS have a stage of prediction bias cancellation. A simple method is used to classify and label the neighborhood of the coded pixel, the label is named context. For each context the cumulated error is determined and prediction error corrected, see [1], [4]. When optimizing ALCM in [60] it has been discovered that single predictor bias cancelling formula can be unreliable, the fact already suggested in [48]. That is why in our methods we are computing the weighted sum of outputs from several bias canceling methods, detailed description of the most advanced 12-component one can be found in [16].

III. The New Predictor Blending Algorithm

The algorithm is a predictor blending technique, which means that it consists in computation of a sample estimate as a weighted sum of estimates given by subpredictors. The algorithms can be outlined as follows, see also Figure 2:

Algorithm 1: For each image pixel do:

1) Find pixel estimate for each sub-predictor.
1a. Improve it in the next cascade stage by NLMS algorithms.
1b. Remove accumulated bias in the last cascade stage.
2) Do blending of pixel estimates, see e.g. section III-B.

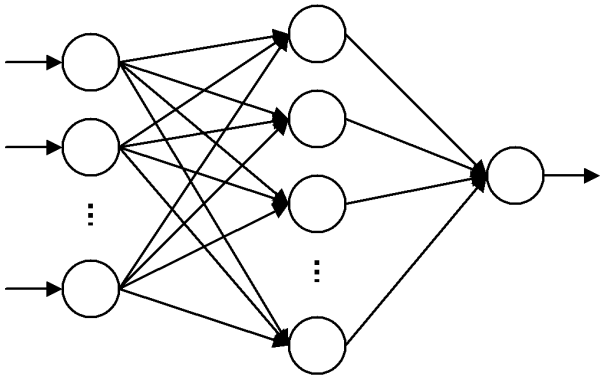

Fig. 3. Network of calculations in Blend-ν algorithms presented in the form similar to that of Multilayer Perceptron network, ν - number of hidden nodes equal to the number of sub-predictors.

3) Code the resultant prediction error by an entropy coder.

Steps 1a. and 1b. are optional. Calculations in step 1 and 2 can be visualized by a network similar to a Multilayer Perceptron one, Figure 3: input nodes keep pixel values from the coded pixel surrounding, Figure 1, then follow nodes representing sub-predictors, section III-A, which outputs meet in a predictor blending node, section III-B.

Reasonably chosen subpredictors should have complementary properties. Another critical factor is the prediction blending formula, possibly adaptive, as the signal is non-stationary one. The proposed method is a "brute-force" approach, as the set of predictors is large (27, hence it's name Blend-27), and some of them are powerful even when used alone, section III-A. A highly sophisticated blending formula is a result of nearly twenty years of experiments, section III-B. High performance entropy coder is described in [16].

A. Sub-predictors

A predictor could perform very well in one image area and could fail in others [71]. Majority of image data consist of edges, textures, regions where intensity varies smoothly, and variable amount of noise [72]. If low implementation complexity is required, simple fixed sub-predictors of order from 1 to 3 are applied [7]. For a sub-predictor of order 1 the estimate is a neighboring pixel value, the pixels $P(i)$ chosen here are for $i = \{1, 2, 3, 4, 5, 6, 10, 13, 18, 28\}$, see Fig.1 for their location. Widely used slightly more complex sub-predictors are [72]: GradWest = $2P(1)–P(5)$, GradNorth = $2P(2)–P(6)$, Plane = $P(1) + P(2)–P(3)$, Plane2 = $P(1)–P(2) + P(4)$, GradNW = $2P(3)–P(11)$.

Predictors that use only up to 3 pixels, are simple, but their ability to model complex data such as areas with rich texture or edges is limited [71]. Then, the presented here algorithms also use more complex sub-predictors, the first group being direct implementations of concepts from linear adaptive prediction

TABLE I
SUB-PREDICTORS AND THEIR KEY PARAMETERS OF BLEND-20, AND BLEND-27 METHODS.

Sub-predictors	Blend-20 [20]			Blend-27			
	α_i	m_i	NLMS+	α_i	m_i	NLMS+	C_{mix}
$P(1)$	1.0	6	No	2.0	6	No	No
$P(2)$	1.0	6	No	1.0	6	No	No
$P(3)$	1.0	20	No	1.0	20	No	No
$P(4)$	1.0	16	No	1.0	16	No	No
$P(5)$	1.0	26	No	1.0	26	No	No
$P(6)$	1.0	26	No	1.0	26	No	No
$P(10)$	1.0	14	No	1.0	14	No	No
$P(18)$	1.0	24	No	1.0	24	No	No
$P(28)$	1.0	28	No	1.0	28	No	Yes
$P(13)$	1.0	28	No	1.0	28	No	No
Plane	1.0	20	No	0.5	20	No	No
Plane2	1.5	32	Yes	1.5	32	Yes	No
GradWest	1.5	36	Yes	1.5	36	Yes	No
GradNorth	2.0	32	No	2.0	32	No	No
$2P(3) - P(11)$	1.0	22	No	1.0	22	No	No
TCM/TCM+	1.5	30	No	1.5	24	No	No
TCM2+	-	-	-	2.5	30	No	No
ALCM+	1.5	32	Yes	1.0	32	Yes	Yes
CoBALP+	2.0	36	Yes	2.0	36	Yes	Yes
RLS+	2.0	32	Yes	1.5	32	Yes	Yes
OLS	4.5	40	Yes	4.0	40	Yes	Yes
AVE-WLS1	-	-	-	2.5	40	Yes	No
AVE-WLS2	-	-	-	1.5	36	Yes	No
AVE-WLS1	-	-	-	2.0	36	Yes	Yes
AVE-WLS2	-	-	-	2.0	36	Yes	Yes
RLS2+	-	-	-	2.0	40	Yes	Yes
ALCM2+	-	-	-	1.0	30	Yes	Yes

theory: enhanced versions of ALCM [60], CoBALP [61], RLS [62], all denoted by subscript "+" in Table I, and OLS [65]. The first three have relatively low computational complexity, RLS and OLS consist in constructing autoregressive model of the coded pixel neighborhood, which is justified by the fact that image data tend to be locally stationary [71]. Then an improved version of texture context matching predictor [73] is used, denoted as TCM+, and presented in [74] (in Blend-20 the original TCM is implemented). ALCM2+, and RLS2+ are acting not on pixels, but on their differences defined as in CoBALP [17]. TCM2+ has differently defined pattern size, $m = 5$ instead of $m = 22$ in TCM+ [20]. The remaining four Blend-27 sub-predictors are versions of probably the most efficient adaptive LS method: AVE-WLS1 is described in [16], [75], AVE-WLS2 is identical except for a different image data weighting function based on formula from [76]. The whole list of sub-predictors used in Blend-20 and -27 methods is given in Table I.

B. Blending of sub-predictors

As mentioned above, efficient predictor blending formula should be adaptive, and should favor locally the best sub-predictors providing the smallest prediction errors in a neighborhood. First formulae were introduced in [71], [72], the presented in this paper ones are much more advanced. For example, neighborhood sizes in (5) m_i are now variable, see Table I. The most sophisticated predictor blending formula is used in Blend-27: prediction is obtained by combining results of four blending methods, two of them are based on those for Blend-20 [20].

Let us start with blending method of much simpler Blend-20 algorithm. The first is similar to classical approach implemented in [71], [74], [77]. The used also in Blend-27 neighborhood penalty term $E_i^{(1)}$ for the i-th sub-predictor is:

$$E_i^{(1)} = 1 + \sum_{j=1}^{m_i} \overline{d}_j \cdot |e_i(j)|, \tag{5}$$

$$\overline{d}_j = \frac{1}{\sqrt{(\Delta x_j)^2 + (\Delta y_j)^2}}$$

is the inverse of Euclidian distance between pixels $P(j)$ and $P(0)$, m_i is the neighborhood size, and $e_i(j)$ is prediction error obtained j positions "before" the current prediction error, see Fig.1. Then, weights w_i are calculated:

$$w_i = \alpha_i \cdot \left(\frac{\delta_i}{E_i^{(1)}} \right)^{\beta}, \tag{6}$$

importance factors α_i can be found in Table I, $\beta = 4$, and:

$$\delta_i = \sum_{j=1}^{m_i} \overline{d}_j. \tag{7}$$

Weights of sub-predictor coefficients should sum up to 1, hence, they are normalized (ν is the number of sub-predictors in Blend-ν algorithm):

$$a_i = \frac{w_i}{\sum_{j=1}^{\nu} w_j} \tag{8}$$

Finally, formula for the global predictor is:

$$\hat{x}^{(1)} = \sum_{i=1}^{\nu} a_i \cdot \hat{x}_i. \tag{9}$$

Penalty term for the second dominant predictor used in Blend-20 and Blend-27 is (it is based on proposals from [76], [78]):

$$E_i^{(2)} = \exp\left(\frac{\chi}{\delta} \sum_{j=1}^{m=96} \overline{d}_j \cdot |e_i(j)| \right) \tag{10}$$

where:

$$\chi = \frac{9.6}{\sqrt[3]{\overline{\sigma}^2}}$$

and $\overline{\sigma}^2$ is the arithmetic mean of all weighted variances $\tilde{\sigma}^2$ for consecutive image samples:

$$\tilde{\sigma}^2 = \frac{1}{\delta} \sum_{j=1}^{m=10} \overline{d}_j \cdot (P(j) - \overline{p})^2 \tag{11}$$

$\overline{p}$ is the arithmetic mean of $m = 10$ pixels around the currently coded one. The global predictor $\hat{x}^{(2)}$ is computed similarly as the $\hat{x}^{(1)}$ one, except for formula on w_j, which is simpler, compare (20):

$$w_j = \frac{\alpha_i}{E_i^{(2)}}$$

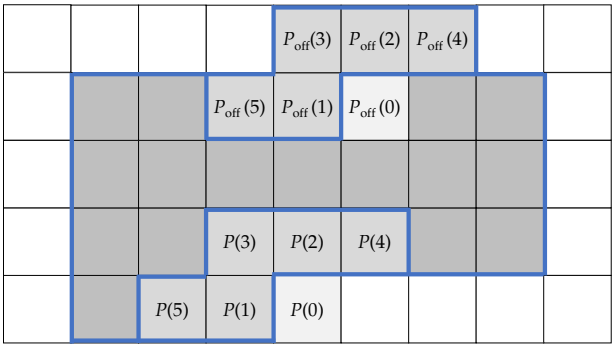

Fig. 4. Search for matching neighborhoods of size m, second starting from $P_{off}(0)$, in training window Q around $P(0)$, $m = 5, W = 3$.

In Blend-20 the final pixel estimate is obtained as a weighted sum of $\hat{x}^{(1)}$ and $\hat{x}^{(2)}$ [20].

The additional penalty terms of Blend-27 are based on a set of previous prediction errors $e^{(i)}_{off(j)}$, the third one is simply:

$$E_i^{(3)} = 1 + \sum_{j=1}^{m_i} |e^{(i)}_{off(j)}|. \tag{12}$$

The $e^{(i)}_{off(j)}$ errors are taken from locations "similar" to that of the coded pixel [79]. Similarity is understood in the same manner as in the TCM+ method [74], [80]: matching neighborhoods are searched, one around currently coded pixel, and one around some other one, $P_{off}(0)$, see Fig.4. The neighborhood is searched for in a training window Q extending W rows above $P(0)$, the rows are $2W + 1$ pixels long and centered around column containing $P(0)$, additionally, W pixels immediately preceding $P(0)$ are included, $W = 14$, Q contains $m = 24$ pixels closest to $P_{off}(0)$. There are 420 neighbourhoods to test, the dissimilarity measure for an l-th neighbourhood is:

$$\Delta_{(l)} = \sum_{k=1}^{m} \overline{d}_k \cdot |P(k) - P_{off(l)}(k)| \tag{13}$$

For each of i predictors 35 previous prediction errors $e^{(i)}_{off(j)}$ are taken from positions for which $\Delta_{(l)}$ parameters are the smallest.

To calculate the fourth penalty term histograms of $e^{(i)}_{off(j)}$ errors are collected, j is histogram index. Using histogram we can evaluate probability of an error value $p(e^{(i)}_{off(j)})$, and hence optimal number of bits needed for its coding equal to $-\log_2 p(e^{(i)}_{off(j)})$. The $E_i^{(4)}$ is then:

$$E_i^{(4)} = -\sum_{j=1}^{K=35} \log_2 p(e^{(i)}_{off(j)}) = K \cdot \log_2 N - \log_2 \prod_{j=1}^{K=35} n_{i,j}, \tag{14}$$

where $n_{i,j}$ is the occurence number of $e^{(i)}_{off(j)}$ value in N error samples used for constructing the histogram.

Knowing all four penalty terms of Blend-27 we can now describe how pixel global estimate is computed. It is obtained as a mean of four "sub-global" estimates $\hat{x}^{(z)}$, $z = 1, 2, 3, 4$:

$$\hat{x}^{(z)} = \sum_{i=1}^{\nu} a_i^{(z)} \cdot \hat{x}_i \tag{15}$$

where similarly as in Blend-20:

$$a_i^{(z)} = \frac{w_i^{(z)}}{\sum_{j=1}^{\nu} w_j^{(z)}}, \tag{16}$$

but calculation of weights is more sophisticated:

$$w_i^{(1)} = \alpha_i \cdot \left(\frac{\delta_i}{E_i^{(1)}}\right)^3 \cdot \left(\frac{1}{E_i^{(3)}}\right)^{1.5}, \tag{17}$$

$$w_i^{(2)} = \alpha_i \cdot \left(\frac{\delta_i}{E_i^{(1)}}\right)^2 \cdot \left(\frac{1}{E_i^{(3)}}\right)^2, \tag{18}$$

$$w_i^{(3)} = \alpha_i \cdot \left(\frac{1}{E_i^{(3)}}\right)^3 \cdot \left(\frac{1}{3 + E_i^{(4)}}\right)^{14}, \tag{19}$$

$$w_i^{(4)} = \frac{\alpha_i}{E_i^{(2)}}. \tag{20}$$

Then, the estimation error is coded by a coder consisting of Golomb and adaptive context arithmetic one, its detailed description can be found [16]. Final coding is preceded by trial runs necessary for optimization of entropy coder settings.

IV. Performance Comparison of State of Art Codecs

A. Note on codecs time complexity

Comparison of programs time complexity written by different authors and run on different equipment is always a tricky problem. Since some time it is even more complicated, as applications can be realized on more than one core (however experimental software usually runs on one), or on GPU, which is even more confusing. Writing an image processing application on a GPU is a problem in itself, however, it usually results in a reduction of execution time, often important. Techniques based on ANNs are in a priviledged position, as there are ready to use applications training ANNs on GPUs. This is probably the reason why declarations concerning time complexity of such codecs are rather enigmatic. Then, in their case it is reasonable to refer to statements from [27], [28] that total computational load for ANN based codecs is usually much higher than for "classical" ones.

It is worth noting that with one exception Blend-27 has smaller computational complexity than its direct competitors, coding and decoding times of Lennagrey image (512×512 pixels) on Pentium i5 3.4 GHz are 292.2 and 254 seconds, respectively. For MRP 0.5 [8] coding time is 420 s, which

means that for its offspring it is even greater: xMRP [9], GPR-BP [11] and MRP-SSP [12]. It should be noted however that they are based on forward prediction, coding process consists in a series of trials, hence decoding time for them is much shorter than coding one. For a very interesting and quite new method PMO 2019 the times are even greater, according to authors of [14] coding and decoding times on Xeon@2.6 GHz are 102 and 18 minutes (512×512 pixel image). It is then very interesting that according to [15] coding time for the newer PMO 2022 is drastically reduced and roughly two times shorter than for Blend-27. On the opposite side are JPEG, CALIC [4], and WebP algorithms coding such image in milliseconds, but generating greater compressed files, output files for JPEG-LS are on average 14.4% greater than for Blend-27 (table II and III). Somwhere between theses extremes are moderately complex coders using backward prediction (hence, having symmetric coding and decoding complexity). Coding times for some of them are: CoBALP$_{ultra2}$ (2.62 s), LA-OLS (5.86 s), GLICBAWLS (7.1 s), Blend-20 (36.65 s), Vanilc WLS-D (52.5 s) and Extedned Multi-WLS (EM-WLS, 107.4 s) [16].

B. Results for lossless coding

Table III contains average numbers of bits per pixel needed to losslessly compres standard image test set ISO/IEC 10918-1. The images are of size 720×576-pixels. Another popular set of 13 images was presented in paper [5], unfortunately not all authors use all of them, hence, results for two partial sets are summarized in Tables IV and V; Mean No 1 is obtained for images: Camera, Couple256, Airplane, Baboon, Lennagrey, Peppers, Ballon, Barb, Barb2, and Gold. First two are of size 256×256-pixels, then come four of size 512×512-pixels, and four of size 720×576-pixels. Mean No 2 is aimed at testing two advanced ANN based codecs: Pixel CNN++, and PMA CNN, Noisesquare and Shapes are added, and Lennagrey omitted, the first is of size 256×256-pixels, and second of size 512×512-pixels. Finally, Table VI is inspired by paper [68], where an overview of ANN codecs is done.

The compared algorithms varies: from very fast and most popular but not particularly efficient from the data compaction point of view, like JPEG-LS, JPEG-2000, and WebP lossless 1.3, to the most efficient, but computationally complex. It is quite clear that usually the best results are obtained for the introduced in this paper Blend-27. Except for one result for GPR-BP ("Airplane"), better results for few images are obtained for PMO 2022 [15], and two ANN-based codecs: PixelCNN++ [25], and PMA CNN [26]. What is characteristic, ANN-based codecs are best for small-resolution 256×256-pixel images. On the other hand, both PMO codecs are particularly good in coding the artificially generated image Shapes. Finally, Blend-27 is unmatched in coding of the largest images.

TABLE II
AVERAGE BIT PER PIXEL RATES FOR NEAR-LOSSLESS ALGORITHMS FOR $d = 2$, AND $d = 10$.

Image	$d = 2$			$d = 10$		
	LOCO-I v.0.90N	TMW mode 3	Blend -27	LOCO-I v.0.90N	TMW mode 4	Blend -27
Bridge256	3.49	3.38	3.205	1.73	1.63	1.469
Camera256	2.28	2.08	1.820	0.96	0.90	0.675
Couple256	1.82	1.60	1.439	0.86	0.70	0.466
Airplane	1.84	1.64	1.495	0.72	0.57	0.448
Baboon	3.72	3.49	3.322	1.91	1.68	1.518
Lennagrey	2.09	1.83	1.699	0.93	0.55	0.408
Peppers	2.29	2.09	1.905	0.93	0.64	0.463
Shapes	0.79	0.75	0.363	0.47	0.58	0.194
Balloon	1.242	0.90	0.733	0.49	0.32	0.169
Boats	1.902	1.65	1.426	0.78	0.66	0.430
Gold	2.333	2.19	2.004	0.99	0.81	0.614
Mean	2.163	1.964	1.765	0.979	0.822	0.623

C. Near-lossless codecs

In Table II comparison of LOCO-I [1], TMW [81], and Blend-27 for near-lossless coding are presented, $d = 2, 10$. Table VII shows few other codecs performance for $d = 1$. In the latter case Blend-27 produces output files shorter by 18.46% than JPEG-LS. In Figure 5 an informal comparison of Blend-27 and WebP 1.3 is done, as WebP is here a fully lossy codec (no limit on maximum error). The figure is based on Table VIII. The determinating parameter is the same value of bitstream generated by the codecs. It can be seen that up to $d = 2$ images coded by Blend-27 have higher PSNR. On the other hand, WebP 1.3 is in this respect better than JPEG and JPEG-2000.

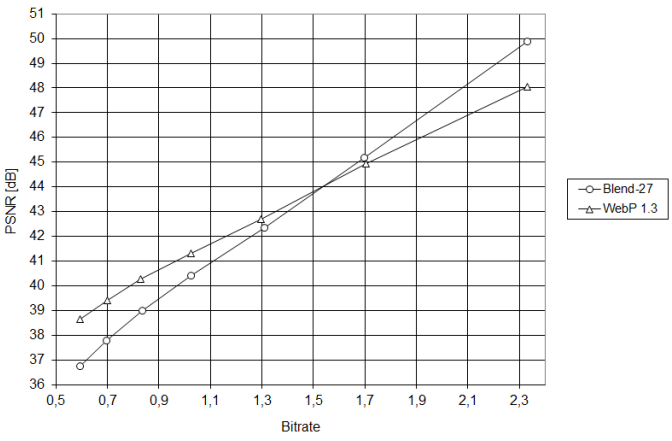

Fig. 5. Comparison of PSNR of coded Lennagray image for near-lossless Blend-27 and WebP 1.3. Circles and triangles represent bitrate/PSNR values taken from Table VIII.

TABLE III
AVERAGE BIT PER PIXEL RATES FOR THE FIRST GROUP OF STANDARD TEST IMAGES. THE BEST RESULTS ARE IN BOLD.

Method	Balloon	Barb	Barb2	Board	Boats	Girl	Gold	Hotel	Zelda	Mean
PNG-crush [82]	3.253	5.214	5.147	3.982	4.287	4.314	4.677	4.809	4.128	4.423
WebP 1.3 [3]	2.961	4.690	4.693	3.730	4.003	3.975	4.495	4.491	3.809	4.094
Sunset [50]	2.89	4.68	4.77	3.73	4.01	3.91	4.56	4.46	3.81	4.091
JPEG-LS [1]	2.889	4.589	4.684	3.674	3.930	3.922	4.475	4.378	3.884	4.058
UCM [50]	2.81	4.44	4.57	3.57	3.85	3.81	4.45	4.28	3.80	3.889
HBB [83]	2.80	4.28	4.48	3.54	3.80	3.74	4.37	4.27	3.72	3.889
CoBALP$_{\max}$ [17]	2.853	4.176	4.440	3.492	3.780	3.696	4.382	4.219	3.749	3.865
CALIC [4]	2.78	4.31	4.46	3.51	3.78	3.72	4.35	4.18	3.69	3.864
CBPC [79]	2.78	4.14	4.47	3.49	3.78	3.70	4.38	4.23	3.72	3.854
Lee [76]	2.79	4.20	4.47	3.50	3.76	3.70	4.35	4.24	3.68	3.854
P13 [71]	2.74	4.29	4.47	3.48	3.75	3.67	4.33	4.19	3.68	3.844
Multi-ctx [49]	2.727	4.243	4.421	3.467	3.730	3.664	4.310	4.171	3.700	3.826
LAT-RLMS [22]	2.75	4.15	4.45	3.48	3.74	3.68	4.34	4.21	3.61	3.823
ALPC [84]	2.74	4.00	4.41	3.48	3.77	3.68	4.39	4.33	3.60	3.822
APC-MAP10 [85]	2.73	4.21	4.45	3.43	3.72	3.62	4.34	4.15	3.64	3.810
AdNN [22]	2.79	4.03	4.45	3.46	3.72	3.63	4.34	4.19	3.61	3.802
APC-A [71]	2.73	4.04	4.40	3.43	3.70	3.61	4.30	4.15	3.63	3.777
OLS [7]	2.690	3.939	4.310	3.388	3.638	3.576	4.273	4.162	3.549	3.725
GLICBAWLS [18]	2.640	3.916	4.318	3.392	3.628	3.565	4.276	4.177	3.537	3.717
BMF [8]	2.649	3.959	4.276	3.331	3.593	3.517	4.238	4.066	3.549	3.686
AdNN$_+$ [24]	2.647	3.868	4.283	3.330	3.607	3.528	4.238	4.086	3.530	3.680
CoBALP$_{\mathrm{ultra2}}$ [61]	2.673	3.881	4.247	3.339	3.591	3.523	4.232	4.067	3.568	3.665
Vanilc WLS-D [86]	2.626	3.815	4.231	3.332	3.589	3.523	4.229	4.074	3.501	3.658
TMW$^{\mathrm{LEGO}}$ [6]	2.60	3.84	4.24	3.27	3.53	3.47	4.22	4.01	3.50	3.631
LA-OLS [16]	2.576	3.832	4.214	3.288	3.537	3.467	4.198	4.040	3.499	3.628
MRP 0.5 [8]	2.579	3.815	4.216	3.268	3.536	3.465	4.207	4.026	3.495	3.623
APC-WLS [7]	2.60	3.75	4.18	3.27	3.53	3.45	4.20	4.01	3.51	3.611
Blend-20 [20]	2.566	3.768	4.175	3.272	3.520	3.449	4.185	4.007	3.498	3.605
EM-WLS [16]	2.546	3.705	4.126	3.240	3.494	3.409	4.169	3.977	3.483	3.572
Blend-27	**2.529**	**3.668**	**4.082**	**3.213**	**3.465**	**3.386**	**4.140**	**3.942**	**3.469**	**3.544**

TABLE IV
AVERAGE BIT PER PIXEL RATES FOR THE SECOND GROUP OF STANDARD TEST IMAGES, PART 1.

Images	JPEG 2000 [2]	WebP lossless 1.3 [3]	FLIF 0.3 [13]	TMW [5]	PMO 2018 [13]	GLIC BAWLS [18]	Vanilc WLS-D [86]	xMRP [9]	MRP 0.5 [8]
Camera	4.535	4.329	4.285	4.098	3.960	4.208	3.995	3.971	3.949
Couple256	3.915	3.727	3.677	3.446	3.415	3.543	3.459	3.389	3.388
Noisesquare	-	-	-	5.542	-	-	5.159	5.301	5.270
Airplane	4.013	3.936	3.794	3.601	3.632	3.668	3.575	3.590	3.591
Baboon	6.107	5.899	6.078	5.738	5.727	5.666	5.678	5.662	5.663
Lennagrey	4.303	4.158	4.252	3.908	3.944	3.901	3.856	3.885	3.889
Peppers	4.629	4.512	4.595	4.251	4.267	4.246	4.187	4.208	4.199
Shapes	-	-	-	0.740	-	-	1.302	0.769	0.685
Balloon	3.031	2.961	2.856	2.649	2.673	2.640	2.626	2.613	2.579
Barb	4.600	4.589	4.500	4.084	3.997	3.916	3.815	3.817	3.815
Barb2	4.789	4.693	4.656	4.378	4.287	4.318	4.231	4.226	4.216
Gold	4.603	4.495	4.518	4.266	4.476	4.276	4.229	4.216	4.207
Mean No 1	4.453	4.330	4.321	4.042	4.038	4.038	3.965	3.958	3.950
Mean No 2	-	-	-	3.890	-	-	3.841	3.797	3.778

TABLE V
AVERAGE BIT PER PIXEL RATES FOR THE SECOND GROUP OF STANDARD TEST IMAGES, PART 2. THE BEST RESULTS ARE IN BOLD.

Images	LA-OLS [16]	BMF [8]	GPR-BP [11]	MRP-SSP [12]	Pixel CNN++ [25]	PMA CNN [26]	PMO 2019 [14]	EM-WLS [16]	PMO 2022 [15]	Blend -27
Camera	4.001	3.952	3.964	3.901	3.749	**3.748**	3.833	3.920	3.804	3.871
Couple256	3.414	3.275	3.339	3.323	**3.176**	**3.176**	3.281	3.345	3.269	3.310
Noisesquare	5.194	-	-	-	5.387	5.375	5.296	5.167	5.274	**5.118**
Airplane	3.568	3.535	**3.451**	3.536	3.486	3.481	3.546	3.547	3.529	3.522
Baboon	5.643	5.677	5.641	5.635	5.698	5.610	5.608	5.622	5.611	**5.604**
Lennagrey	3.881	3.863	3.880	3.877	-	-	3.845	3.847	**3.825**	3.836
Peppers	4.153	4.177	4.170	4.163	4.194	4.192	4.176	4.101	4.161	**4.090**
Shapes	1.109	-	-	-	0.747	0.720	0.497	0.903	**0.490**	0.700
Balloon	2.576	2.649	2.544	2.548	2.579	2.573	2.584	2.546	2.573	**2.529**
Barb	3.832	3.804	3.821	3.764	3.914	3.905	3.733	3.705	3.708	**3.668**
Barb2	4.214	4.163	4.184	4.175	4.270	4.266	4.146	4.126	4.122	**4.082**
Gold	4.198	4.179	4.178	4.173	4.170	4.166	4.191	4.170	4.171	**4.140**
Mean No 1	3.948	3.927	3.917	3.910	-	-	3.903	3.893	3.877	**3.865**
Mean No 2	3.809	-	-	-	3.753	3.746	3.726	3.740	3.701	**3.694**

TABLE VI
AVERAGE BIT PER PIXEL RATES FOR THE THIRD GROUP OF STANDARD TEST IMAGES [68]. THE BEST RESULTS ARE IN BOLD.

Images	BPG	PNG	LCIC	JPEG 2000	JPEG -LS	JPEG -XL	FLIF	WebP	L3C	CWP LIC	LCIC duplex	Blend -27
Airplane	4.32	4.26	3.99	4.00	3.80	3.71	3.82	3.87	4.56	3.69	3.69	**3.495**
Barbara	5.06	5.22	4.61	4.61	4.70	4.40	4.56	4.55	5.44	4.35	4.36	**3.674**
Coastguard	5.70	5.06	4.82	4.83	4.86	4.73	4.93	4.81	5.82	4.80	4.83	**4.265**
Comic	6.15	5.84	5.63	5.65	5.30	5.07	5.50	5.45	6.60	4.83	4.83	**4.661**
Flowers	5.18	5.08	4.91	4.92	4.62	4.51	4.74	4.76	5.53	4.41	4.35	**4.217**
Goldhill	4.95	4.70	4.58	4.59	4.43	4.37	4.50	4.47	5.27	4.33	4.33	**4.084**
Lennagrey	4.54	4.61	4.31	4.31	4.24	4.16	4.28	4.14	4.95	4.13	4.08	**3.836**
Mandrill	6.61	6.23	6.11	6.11	6.04	5.98	6.14	5.89	6.97	5.95	5.89	**5.606**
Monarch	4.10	4.26	3.82	3.82	3.70	3.54	3.68	3.73	4.37	3.40	3.45	**3.259**
Pepper	4.77	4.90	4.63	4.63	4.51	4.48	4.58	4.50	5.38	4.67	4.38	**4.083**
Ppt3	2.20	2.35	2.41	2.41	2.04	1.84	1.87	2.06	3.71	2.14	2.07	**1.635**
Zebra	5.83	5.19	4.89	4.89	4.81	4.66	4.84	4.86	6.08	4.65	4.68	**4.251**
Mean	4.951	4.808	4.559	4.564	4.421	4.288	4.453	4.424	5.390	4.279	4.245	**3.922**

TABLE VII
AVERAGE BIT PER PIXEL RATES FOR NEAR-LOSSLESS ALGORITHMS FOR $d = 1$. THE BEST RESULTS ARE IN BOLD.

Images	JPEG -LS [1]	Sunset [34]	BAROLTO [34]	ASBOSC [31]	CALIC [4]	PNLIC [32]	GLICBA WLS [18]	BMF	Blend -27
Balloon	1.465	1.45	1.48	1.457	1.461	1.554	1.321	1.323	**1.239**
Barb1	3.149	3.10	2.88	2.868	2.837	3.093	2.428	2.475	**2.204**
Barb2	3.174	3.17	3.06	3.012	2.986	3.258	2.813	2.783	**2.602**
Board	2.203	2.22	2.09	2.033	2.038	2.234	1.938	1.884	**1.784**
Boats	2.478	2.48	2.39	2.306	2.315	2.554	2.168	2.139	**2.022**
Girl	2.446	2.38	2.28	2.258	2.250	2.500	2.107	2.070	**1.954**
Gold	2.996	3.06	2.92	2.904	2.874	3.080	2.754	2.740	**2.648**
Hotel	2.873	2.94	2.75	2.719	2.697	3.043	2.660	2.559	**2.438**
Zelda	2.375	2.26	2.25	2.221	2.225	2.264	2.048	2.070	**1.993**
Mean	2.573	2.56	2.46	2.420	2.409	2.260	2.249	2.227	**2.098**

TABLE VIII
COMPARISON OF PSNR OF NEAR-LOSSLESS CODED LENNAGRAY IMAGE USING BLEND-27 TO PSNR OBTAINED FOR WEBP 1.3 FOR APPROXIMATELY THE SAME BITRATES.

d	Blend-27		WebP 1.3	
	Bitrate	PSNR [dB]	Bitrate	PSNR dB]
1	2.330	49.884	2.330	48.04
2	1.699	45.178	1.704	44.94
3	1.310	42.335	1.298	42.70
4	1.025	40.426	1.025	41.30
5	0.835	38.977	0.831	40.27
6	0.697	37.770	0.701	39.41
7	0.593	36.745	0.593	38.65

V. CONCLUSION

A new highly efficient predictor blending algorithm is described in the paper, Blend-27. Tests show that it is probably the best method for losslessly coding of images, if the data compaction property is taken into consideration. This is the conclusion of an extensive overview of existing methods performance, including that of the newest algorithms based on Artificial Neural Networks. Efficiency of the new technique as a near-lossless codec is also considered. At the same time with one exception its computational complexity is smaller than that of its direct competitors.

REFERENCES

[1] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000. [Online]. Available: https://doi.org/10.1109/83.855427

[2] M. Marcellin, M. Gormish, A. Bilgin, and M. Boliek, "An overview of JPEG-2000," in *Proceedings DCC 2000. Data Compression Conference*, 2000, pp. 523–541. [Online]. Available: https://doi.org/10.1109/DCC.2000.838192

[3] C. W. 1.3. (loaded 2023-04-08). [Online]. Available: https://storage.googleapis.com/downloads.webmproject.org/releases/webp/libwebp-1.3.0-windows-x64.zip

[4] X. Wu and N. Memon, "CALIC - a context based adaptive lossless image codec," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 4, 1996, pp. 1890–1893 vol. 4. [Online]. Available: https://doi.org/10.1109/ICASSP.1996.544819

[5] B. Meyer and P. Tischer, "TMW – a new method for lossless image compression," in *Proceedings of International Picture Coding Symposium (PCS97)*, 1997, pp. 533–538.

[6] ——, "TMW^{Lego} - an object oriented image modeling framework," in *Proceedings of Data Compression Conference*, 2001, p. 504.

[7] H. Ye, G. Deng, and J. Devlin, "A weighted least squares method for adaptive prediction in lossless image compression," in *Picture Coding Symp. PCS'03*, 2003, pp. 489–493.

[8] I. Matsuda, N. Ozaki, Y. Umezu, and S. Itoh, "Lossless coding using variable blok-size adaptive prediction optimized for each image," in *Proceedings of 13th European Signal Processing Conference EUSIPCO-05 CD*, 2005.

[9] F.-Y. Hsieh, C.-M. Wang, C.-C. Lee, and K.-C. Fan, "A lossless image coder integrating predictors and block-adaptive prediction," *Journal of Information Science and Engineering*, vol. 24, no. 5, pp. 1579–1591, 2008.

[10] X. Wu, G. Zhai, X. Yang, and W. Zhang, "Adaptive sequential prediction of multidimensional signals with applications to lossless image coding," *IEEE Trans. on Image Proces.*, vol. 20, no. 1, pp. 36–42, 2011.

[11] W. Dai and H. Xiong, "Gaussian process regression based prediction for lossless image coding," in *Proceedings of Data Compression Conference*, 2014, pp. 93–102.

[12] W. Dai, H. Xiong, J. Wang, and Y. Zheng, "Large discriminative structured set prediction modeling with max-margin markov network for lossless image coding," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 541–554, 2014.

[13] I. Matsuda, N. Ozaki, Y. Umezu, and S. Itoh, "A lossless image coding method based on probability model optimization," in *36th Int. Conf. Acoustics, Speech and Signal Proces. ICASSP'11*, 2018, pp. 156–160.

[14] K. Unno, Y. Kameda, I. Matsuda, S. Itoh, and S. Naito, "Lossless image coding exploiting local and non-local information via probability model optimization," in *27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[15] H. K. et al., "Improved probability modeling for lossless image coding using example search and adaptive prediction," in *IWAIT2022*, 2022.

[16] G. Ulacha, R. Stasinski, and C. Wernik, "Extended multi WLS method for lossless image coding," *Entropy*, vol. 22, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/9/919

[17] T. Strutz, "Context-based adaptive linear prediction for lossless image coding," in *Proceedings of the 4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, jan 2002, pp. 105–109.

[18] B. Meyer and P. Tischer, "GLICBAWLS - grey level image compression by adaptive weighted least squares," in *Proceedings of Data Compression Conference*, 2000, p. 503.

[19] L.-J. Kau, Y.-P. Lin, and C.-T. Lin, "Lossless image coding using adaptive, switching algorithm with automatic fuzzy context modelling," *Vision, Image and Signal Processing, IEE Proceedings -*, vol. 153, pp. 684 – 694, 11 2006. [Online]. Available: https://doi.org/10.1049/ip-vis:20045256

[20] G. Ulacha and R. Stasiński, "Performance optimized predictor blending technique for lossless image coding," in *36th Int. Conf. Acoustics, Speech and Signal Proces. ICASSP'11*, 2011, pp. 1541–1544.

[21] S. Marusic and G. Deng, "A neural network based adaptive non-linear lossless predictive coding technique," in *ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359)*, vol. 2, 1999, pp. 653–656 vol.2. [Online]. Available: https://doi.org/10.1109/ISSPA.1999.815757

[22] ——, "Adaptive prediction for lossless image compression," *Signal Processing: Image Communication*, vol. 17, no. 5, pp. 363–372, 2002. [Online]. Available: https://doi.org/10.1016/S0923-5965(02)00006-1

[23] K. Takizawa, S. Takenouchi, H. Aomori, T. Otake, M. Tanaka, I. Matsuda, and S. Itoh, "Lossless image coding by cellular neural networks with minimum coding rate learning," in *2011 20th European Conference on Circuit Theory and Design (ECCTD)*, 2011, pp. 33–36. [Online]. Available: https://doi.org/10.1109/ECCTD.2011.6043337

[24] G. Ulacha and R. Stasiński, "Improving neural network approach to lossless image coding," in *Proceedings of The 29th Picture Coding Symposium PCS'12*, 2012, pp. 173–176.

[25] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," in *Proc. 5th International Conference on Learning Representations (ICLR 2017)*, 2017.

[26] H. Kojima, Y. Kameda, Y. Kita, I. Matsuda, and S. Itoh, "Probability model adjustment for the CNN-based lossless image coding method," in *Proc. SPIE 11766, International Workshop on Advanced Imaging Technology (IWAIT) 2021*, 2021.

[27] S. Zhang, C. Zhang, N. Kang, and Z. Li, "iVPF: Numerical invertible volume preserving flow for efficient lossless compression," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1–10.

[28] Y. Bai, X. Liu, K. Wang, X. Ji, X. Wu, and W. Gao, "Deep lossy plus residual coding for lossless and near-lossless image compression," *arXiv - CS - Computer Vision and Pattern Recognition*, 2022.

[29] G. Ulacha and R. Stasinski, "High performance predictor blending lossless image coder," in *Data Compression Conference 2023 (DCC)*, 2023, p. 366.

[30] H. Hartenstein, R. Herz, and D. Saupe, "A comparative study of L_∞-distortion limited image compression algorithms," in *Proceedings of International Conference on Image Processing ICIP'03*, 2003, pp. 1–5.

[31] R. Iordache, I. Tabus, and J. Astola, "Fixed-slope near-lossless context-based image compression," in *Proceedings of 1998 International Conference on Image Processing*, vol. 1, 1998, pp. 512–515.

[32] A. Krivoulets, "A method for progressive near-lossless image compression," in *Proceedings of Picture Coding Symposium*, vol. 2, 1997, pp. 185–188.

[33] X. Xie, G. Li, D. Li, C. Zhang, and Z. H. Wang, "A new near-lossless image compression algorithm suitable for hardware design in wireless endoscopy system," in *IEEE International Conference on Image Processing 2005*, vol. 1, 2005, pp. I–1125. [Online]. Available: https://doi.org/10.1109/ICIP.2005.1529953

[34] X. Xue, "Prediction based on backward adaptive recognition of local texture orientation and poisson statistical model for lossless/near-lossless image compression," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, 1999, pp. 3137–3140.

[35] J. Ström and P. C. Cosman, "Medical image compression with lossless regions of interest," *Signal Processing*, vol. 59, no. 2, pp. 155–171, 1997, biomedical Imaging.

[36] Y. Kuroki, Y. Ueshige, and T. Ohta, "An estimation of the predictors implemented by shift operation, addition, and/or substraction," in *Proceedings of International Conference on Image Processing 2001*, 2001, pp. 474–477.

[37] K. Sayood, Ed., *Introduction to Data Compression*, 5th ed. Morgan Kaufmann, 2018.

[38] J. Jiang and C. Grecos, "Towards an improvement on prediction accuracy in JPEG-LS," *Optical Engineering*, vol. 41, no. 2, pp. 335 – 341, 2002. [Online]. Available: https://doi.org/10.1117/1.1428743

[39] H. Wang and D. Zhang, "A linear edge model and its application in lossless image coding," *Signal Processing: Image Communication*, vol. 19, no. 10, pp. 955–958, 2004.

[40] G. Ulacha and R. Stasinski, "On context-based predictive techniques for lossless image compression," *IWSSIP 2005 - Proceedings of 12th International Workshop on Systems, Signals and Image Processing*, pp. 345–348, 11 2005.

[41] A. Avramović, "Lossless compression of medical images based on gradient edge detection," in *2011 19thTelecommunications Forum (TELFOR) Proceedings of Papers*, 2011, pp. 1199–1202. [Online]. Available: https://doi.org/10.1109/TELFOR.2011.6143765

[42] A. Attar, R. M. Rad, and A. Shahbahrami, "An accurate gradient-based predictive algorithm for image compression," in *MoMM '10: Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, 2010, p. 374–377.

[43] C.-C. Chang and G.-I. Chen, "Enhancement algorithm for nonlinear context-based predictors," *Vision, Image and Signal Processing, IEE Proceedings* -, vol. 150, pp. 15–19, 03 2003. [Online]. Available: https://doi.org/10.1049/ip-vis:20030163

[44] A. Seyed Danesh, R. Moradi Rad, and A. Attar, "A novel predictor function for lossless image compression," in *2010 2nd International Conference on Advanced Computer Control*, vol. 2, 2010, pp. 527–531. [Online]. Available: https://doi.org/10.1109/ICACC.2010.5486699

[45] D. Estrakh, H. Mitchell, P. Schaefer, Y. Mann, and Y. Peretz, ""Soft" median adaptive predictor for lossless picture compression," *Signal Processing*, vol. 81, no. 9, pp. 1985–1989, 2001. [Online]. Available: https://doi.org/https://doi.org/10.1016/S0165-1684(01)00058-5

[46] A. Itani and M. Das, "Adaptive switching linear predictor for lossless image compression," in *Advances in Visual Computing*, G. Bebis, R. Boyle, D. Koracin, and B. Parvin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 718–722.

[47] N. Karimi, S. Samavi, and S. Shirani, "Lossless compression of high-throughput RNAi images," in *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine*, 2010, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ITAB.2010.5687771

[48] C. Topal and O. N. Gerek, "Pdf sharpening for multichannel predictive coders," in *2006 14th European Signal Processing Conference*, 2006, pp. 1–4.

[49] G. Ulacha and R. Stasinski, "A new fast multi-context method for lossless image coding," in *Proceedings of the 2018 International Conference on Sensors, Signal and Image Processing*, ser. SSIP 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 69–72. [Online]. Available: https://doi.org/10.1145/3290589.3290600

[50] N. D. Memon and K. Sayood, "An asymmetric lossless image compression technique," in *Proceedings of the 1995 International Conference on Image Processing*, vol. 3, 1995, pp. 97–100.

[51] F. Golchin and K. K. Paliwal, "Classified adaptive prediction and entropy coding for lossless coding of images," in *Proceedings of International Conference on Image Processing*, 1997, pp. 110–113.

[52] B. Aiazzi, S. Baronti, and L. Alparone, "Near-lossless image compression by relaxation-labeled prediction," *Signal Processing*, vol. 82, no. 11, pp. 1619–1631, 2002.

[53] M. Salami, M. Iwata, and T. Higuchi, "Lossless image compression by evolvable hardware," in *Fourth European Conference on Artificial Life*, Brighton, UK, 1997, pp. 407–416.

[54] S. Takamura, M. Matsumura, and Y. Yashima, "A study on an evolutionary pixel predictor and its properties," in *Proceedings of the 16th IEEE International Conference on Image Processing*, ser. ICIP'09. IEEE Press, 2009, p. 1901–1904.

[55] Y.-G. Wu, "Differential pulse code modulation predictor design procedure using a genetic algorithm," *Optical Engineering*, vol. 42, no. 6, pp. 1649 – 1655, 2003. [Online]. Available: https://doi.org/10.1117/1.1572889

[56] Y. Hashidume and Y. Morikawa, "Lossless image coding based on minimum mean absolute error predictors," in *SICE Annual Conference 2007*, 2007, pp. 2832–2836. [Online]. Available: https://doi.org/10.1109/SICE.2007.4421471

[57] G. Ulacha and M. Łazoryszczak, "Lossless image coding using non-mmse algorithms to calculate linear prediction coefficients," *Entropy*, vol. 25, no. 1, pp. 1–19, 2023.

[58] N. Boulgouris, S. Zaharos, and M. Strintzis, "Adaptive decorrelation and entropy coding for context-based lossless image compression," in *1st Balkan Conference on Signal Processing, Communications, Circuits, and Systems. Istanbul, Turkey*, 2000.

[59] T. Strutz, "Context-based predictor blending for lossless colour image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 4, pp. 687–695, 2016.

[60] G. Ulacha and R. Stasiński, "A time-effective lossless coder based on hierarchical contexts and adaptive predictors," in *14th IEEE Mediterranean Electrotech. Conf. MELECON'08*, 2008, pp. 829–834.

[61] ——, "New context-based adaptive linear prediction algorithm for lossless image coding," in *Proceedings of Int. Conf. on Signals and Electronic Systems (ICSES'14)*, 2014, pp. 1–4.

[62] G. Ulacha and R. Stasinski, "Context based lossless coder based on rls predictor adaption scheme," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1917–1920. [Online]. Available: https://doi.org/10.1109/ICIP.2009.5413680

[63] X. Wu, E. Barthel, and W. Zhang, "Piecewise 2D autoregression for predictive image coding," in *Proceedings 1998 International Conference*

on Image Processing. ICIP98 (Cat. No.98CB36269), 1998, pp. 901–904 vol.3. [Online]. Available: https://doi.org/10.1109/ICIP.1998.727397

[64] H. Ye, G. Deng, and J. Devlin, "Adaptive linear prediction for lossless coding of greyscale images," in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, vol. 1, 2000, pp. 128–131 vol.1. [Online]. Available: https://doi.org/10.1109/ICIP.2000.900911

[65] ——, "Least squares approach for lossless image coding," in *ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359)*, vol. 1, 1999, pp. 63–66 vol.1. [Online]. Available: https://doi.org/10.1109/ISSPA.1999.818113

[66] F. Mentzer, L. van Gool, and M. Tschannen, "Learning better lossless compression using lossy compression," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6637–6646.

[67] H. Rhee, Y. I. Jang, S. Kim, and N. I. Cho, "LC-FDNet: Learned lossless image compression with frequency decomposition network," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6023–6032.

[68] ——, "Lossless image compression by joint prediction of pixel and context using duplex neural networks," *IEEE Access*, vol. 9, pp. 86 632–86 645, 2021.

[69] E. Hoogeboom, J. W. Peters, R. van den Berg, and M. Welling, "Integer discrete flows and lossless compression," in *Conference on Neural Information Processing Systems (2019)*, 2019.

[70] H. Huang, P. Fränti, D. Huang, and S. Rahardja, "Cascaded RLS-LMS prediction in MPEG-4 lossless audio coding," *IEEE Trans. on Audio, Speech and Language Proces.*, vol. 16, no. 3, pp. 554–562, 2008.

[71] "Transform domain LMS-based adaptive prediction for lossless image coding," *Signal Processing: Image Communication*, vol. 17, no. 2, pp. 219–229, 2002. [Online]. Available: https://doi.org/10.1016/S0923-5965(01)00019-4

[72] T. Seemann and P. Tischer, "Generalized locally adaptive DPCM," *Department of Computer Science Technical Report CS97/301*, pp. 1–15, 1997.

[73] L.-J. Kau and Y.-P. Lin, "Lossless image coding using a switching predictor with run-length encodings," in *IEEE Int. Conf. on Multimedia and Expo*, 2004, pp. 1155–1158.

[74] G. Ulacha and R. Stasiński, "Highly effective predictor blending method for lossless image coding," in *15th IEEE Mediterranean Electrotech. Conf. MELECON'10*, 2010, pp. 1099–1104.

[75] ——, "Enhanced lossless image coding methods based on adaptive predictors," in *Int. Conf. on Systems, Signals and Image Proces. IWSSIP 2010*, 2010, pp. 312–315.

[76] W. S. Lee, "Edge-adaptive prediction for lossless image coding," in *Data Compression Conf. DCC'99*, 1999, pp. 483–490.

[77] G. Ulacha and R. Stasiński, "Improved predictor blending technique for lossless image coding," in *Int. Conf. on Signals and Electronic Systems ICSES'10*, 2010, pp. 115–118.

[78] G. Schuller, B. Yu, and D. Huang, "Lossless coding of audio signals using cascaded prediction," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 2001, pp. 3273–3276.

[79] J. Knezovic and M. Kovac, "Gradient based selective weighting of neighboring pixels for predictive lossless image coding," in *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.*, 2003, pp. 483–488. [Online]. Available: https://doi.org/10.1109/ITI.2003.1225390

[80] G. Ulacha and R. Stasiński, "Texture matching method for lossless image coding," in *International Conference on Systems, Signals and Image Processing IWSSIP'07*, 2007, pp. 135–138.

[81] B. Meyer and P. Tischer, "Extending TMW for near lossless compression of greyscale images," in *Proceedings of Data Compression Conference 1998*, 1998, pp. 458–470.

[82] K. Sayood, *Lossless Compression Handbook*, ser. Communications, Networking and Multimedia. Elsevier Science, 2002.

[83] T. Seemann, P. Tischer, and B. Meyer, "History-based blending of image sub-predictors," in *Proc. Picture Coding Symposium*, 1997, pp. 147–151.

[84] G. Motta, J. A. Storer, and B. Carpentieri, "Improving the performance of adaptive linear prediction coding (ALPC) via least square minimization," in *Proceedings of 12th International Workshop on Systems, Signals and Image Processing - IWSSIP 2005*, 2005, pp. 335–338.

[85] A. Martchenko and G. Deng, "Bayesian predictor combination for lossless image compression," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5263–5270, 2013.

[86] A. Weinlich, P. Amon, A. Hutter, and A. Kaup, "Probability distribution estimation for autoregressive pixel-predictive image coding," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1382–1395, 2016.